

1 Affine Transformations

Thus far we have plotted coupler curves for a single point. But the kangaroo has large hind feet. *How can we plot the swept path of a foot?* Also, as we discovered, the tangent half-angle solution assumes that a local x axis is aligned with the ground link (*link1*). *How can we transform the computed coupler curve to a rotated (X,Y) world frame?*

Affine (shape preserving) transformations are useful whenever you have a rigid body or a cloud of points (e.g., a coupler curve) and you want to compute:

- *new coordinates* after you translate and rotate the body
- coordinates in a *new coordinate frame* translated and rotated with respect to the original frame.

A useful way to handle the transformations is to first convert points (e.g., vertices of a polygon, or points of a coupler curve) to *homogeneous coordinates*. This method is used widely in robotics and computer graphics. We augment each point, (p_x, p_y) , with a '1' to create a column vector $\mathbf{p} = [p_x, p_y, 1]^t$ of homogeneous coordinates (where t signifies transpose).

Then we define 3x3 transformation matrices for translations by $\mathbf{r} = [r_x, r_y]^t$:

$$\begin{bmatrix} p'_x \\ p'_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & r_x \\ 0 & 1 & r_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} \quad \text{or: } \mathbf{p}' = [T]\mathbf{p} \quad (1)$$

and rotations by θ (anti-clockwise, in radians):

$$\begin{bmatrix} p'_x \\ p'_y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} \quad \text{or: } \mathbf{p}' = [R]\mathbf{p} \quad (2)$$

These matrices correspond to the operations shown in Fig. 1.

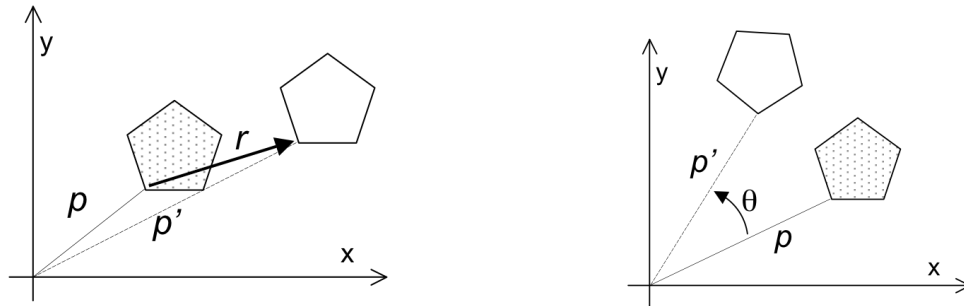


Figure 1: Translation by (r_x, r_y) and rotation by θ

Note that multiplying by T^{-1} is the same as translating by $(-r_x, -r_y)$, and multiplying by R^{-1} is the same as rotating by $-\theta$. Thus, if we keep the body fixed and instead translate or rotate the coordinate frame, we have $\mathbf{p}' = T^{-1}\mathbf{p}$ and $\mathbf{p}' = R^{-1}\mathbf{p}$ using T and R as defined above. The corresponding operations are shown in Fig. 2.

Finally, because translation and rotation are both 3x3 matrices, we can concatenate the results, right to left, to show the effects of sequential translations and rotations. For example, suppose $T1$

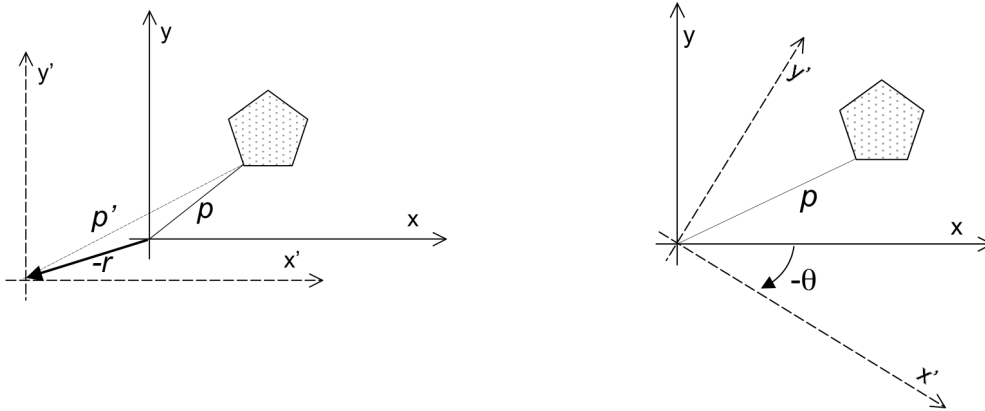


Figure 2: Translating the coordinate frame by $(-r_x, -r_y)$ has the same effect as translating the polygon by (r_x, r_y) . Rotating the coordinate frame by $-\theta$ has the same effect as rotating the polygon by θ about the origin.

is a translation that would move the object to the origin. What does the following achieve?

$$\mathbf{p}' = [T1]^{-1}[R][T1]\mathbf{p} \quad (3)$$

1.1 Transforming polygons or point clouds

Now suppose we have a group of points – for example a coupler curve or a polygon representing a foot. To translate and rotate the polygon, we augment each point with a '1' and put them into the columns of a matrix. Thus, to perform the same operation as above, where we translate to the origin, rotate, and then translate back out again, we have:

$$\begin{bmatrix} p1'_x & p2'_x & p3'_x & p4'_x & \text{etc.} \\ p1'_y & p2'_y & p3'_y & p4'_y & \text{etc.} \\ 1 & 1 & 1 & 1 & \text{etc.} \end{bmatrix} = [T1]^{-1}[R][T1] \begin{bmatrix} p1_x & p2_x & p3_x & p4_x & \text{etc.} \\ p1_y & p2_y & p3_y & p4_y & \text{etc.} \\ 1 & 1 & 1 & 1 & \text{etc.} \end{bmatrix} \quad (4)$$

1.2 Using coupler curves to create polygon trajectories

Now we are ready to read the (x, y, θ) values for each location on a coupler curve and use those values to sweep a polygon through a path in the plane, adjusting its orientation at each point. Note that until now, we have not cared about the orientation of a link which is attached to a coupler point. A point doesn't have orientation, but a foot does!

Let θ give the orientation of a link to which the coupler is attached. For example, with a 4-bar rocker-crank linkage, it is usually $\theta = \theta_3 + \gamma_c$ (see previous class notes for terminology). The code fragment below reads a series of (x, y, θ) points, line by line, from a file and creates translations (x_i, y_i) and rotations, θ_i , for each point. It then applies these transformations to a polygon.

```
#Get coupler curve data, including orientations: x_i, y_i, theta_i    in each row
couplerdata = loadtxt('HalfAngleOutput.txt')
```

```
#Get a polygon (perhaps a foot?) in terms of its vertices: x_j, y_j    in each row
polygon = loadtxt('PointsData.txt')
```

```

#Add ones so each polygon point is now (x,y,1) in homogeneous coords
#Put them all in a matrix, transposed to columns as in equation (4).
rcol = np.ones(len(polygon))
homogeneous = column_stack((polygon,rcol)) #use horzcat( ) in Matlab
oldpoints = np.matrix(transpose(homogeneous))

#If you want a constant "offset" orientation applied to the polygon for
# every location around the coupler curve, you can do it here
#Optional local (constant) reorientation
RotPoly = rotq(1.1*pi/2.)
#Optional global rotation about origin
RotGlobal = rotq(pi)

#Finally, for each coupler point (including angle), transform
#the foot to the desired position and orientation to create a
#sequence of foot positions and orientations.
for i in range(0,len(couplerdata)):
    x = couplerdata[i,0]      #x of i'th coupler point
    y = couplerdata[i,1]      #y of i'th coupler point
    theta = couplerdata[i,2]  #theta of i'th coupler point
    Rot = rotq(theta)
    Tran = tranxy(x,y)
    points = RotGlobal*Tran*RotPoly*Rot*oldpoints #Just multiply them all together!
    plotlines(points,'b','r') #Plot the transposed shape

```

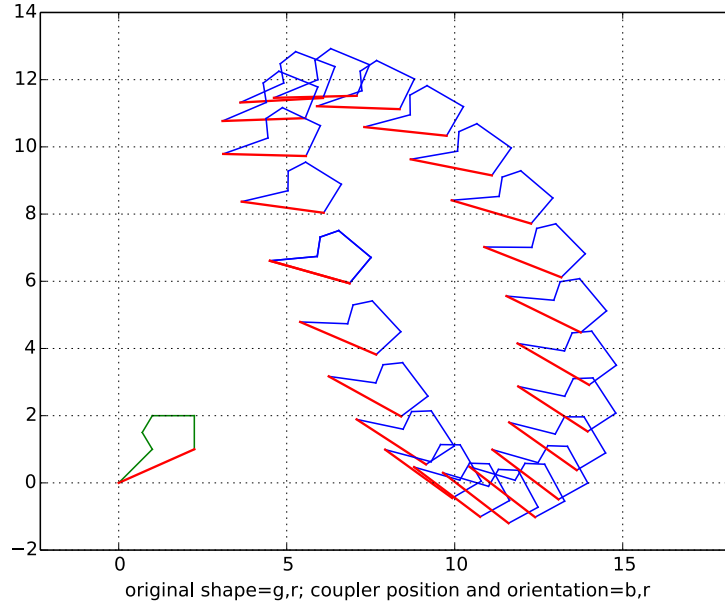


Figure 3: Example of transforming a polygon around a coupler curve

2 Revisiting Power and Virtual Work – for mechanisms

We saw earlier that following power from a motor, through a transmission, etc. could be useful for analysis. For example, for the crawlers: $\tau_L \omega_L - P_{Loss} = f_x v_x$ where τ_L is the motor output torque, P_{Loss} represents various losses, and f_x and v_x are the force and velocity for going up a track.

The same principles apply for mechanisms, but we take the dot product of \vec{F} and velocity \vec{V} at points around the coupler curve. Working from a motor, through a transmission and linkage:

$$\tau_L \omega_L (\eta_{trans} \cdot \eta_{link}) = \vec{F}(t) \cdot \vec{V}(t) \quad (5)$$

where η_{trans} and η_{link} are the efficiencies of the transmission and linkage. Although the motor may run at constant velocity, its torque τ_L varies for each angle, θ_2 , of the mechanism due to variations in \vec{F} and \vec{V} . If we can estimate \vec{F} and \vec{V} , and if we have estimates of our transmission and linkage efficiencies (probably similar to those in the crawlers), we can predict the required motor torque.

2.1 Obtaining the Velocity

We can use graphical, algebraic or numerical methods to get the velocity.

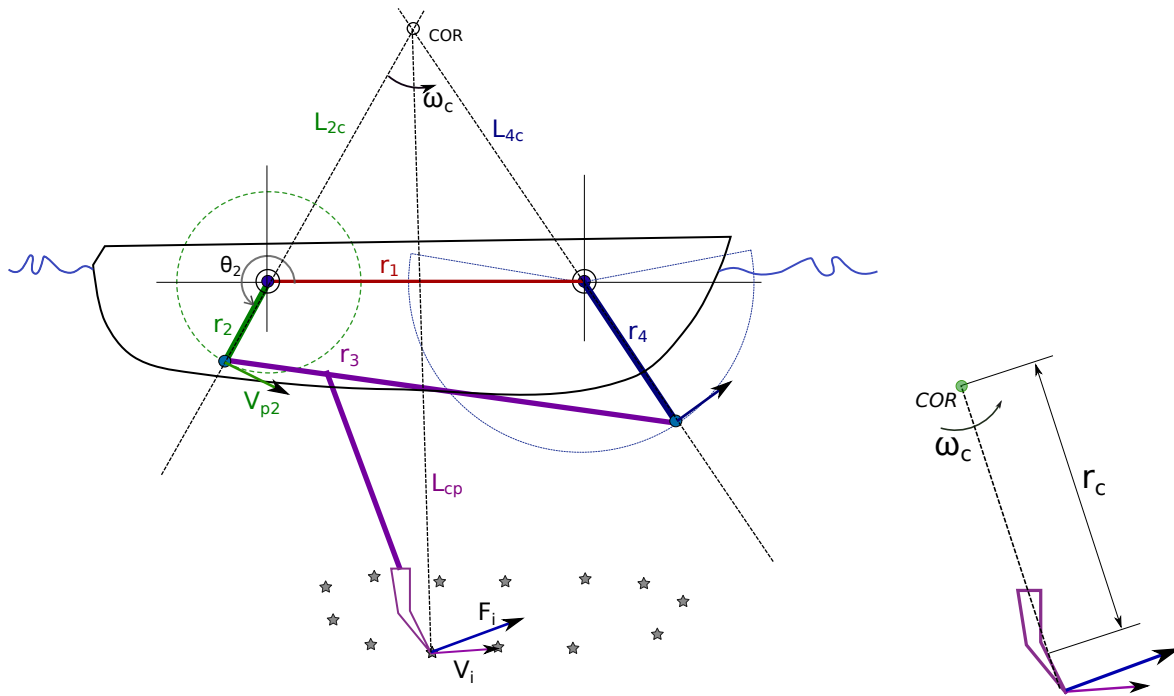


Figure 4: (left) A paddling boat linkage showing instantaneous C.O.R. of coupler point, leading to estimate of \vec{V}_i at the i 'th location of the crank input, θ_{2i} . (right) detail of a paddle rotating about a C.O.R. and showing the radius of gyration r_c .

2.1.1 Graphical velocity estimates

The graphical method is good for a quick estimate of velocity at a few points. Recall that any body translating and rotating in a plane is virtually rotating about some instantaneous *center of rotation*, C.O.R. If we pick two points for which we know the velocity, we can find the C.O.R. and, using it, the velocity of *any other points*. For example, Fig. 4 shows a “duck-bot” from 2014. We can use the C.O.R. to get the velocity of any part of the leg in the water:

- Draw lines L_{2c} and L_{4c} perpendicular to the known motions of the joints between links 2-3 and 3-4.
- The intersection of the two lines is the instantaneous C.O.R. The magnitude of the velocity at the tip of r_2 is also known: $v_{p2} = \omega_2 r_2$, so $\omega_c = v_{p2}/|L_{2c}|$ where $|L_{2c}|$ is the distance from the joint at the tip of r_2 to the C.O.R.
- Draw a line L_{cp} from the C.O.R. to the point of interest.
- The point’s velocity will be perpendicular to L_{cp} and its magnitude will be $V = \omega_c |L_{cp}|$.

2.1.2 Algebraic velocity calculation

We can return to the linkage analysis notes from last week and calculate the velocities ω_3 and v_{p2} directly, knowing ω_2 and the configuration of the linkage for each θ_2 . Differentiating the (x,y) vector loop equations gives two (linear) equations in ω_3 and ω_4 :

$$-r_2 \sin \theta_2 \omega_2 - r_3 \sin \theta_3 \omega_3 + r_4 \sin \theta_4 \omega_4 - r_1 = 0 \quad (6)$$

$$r_2 \cos \theta_2 \omega_2 + r_3 \cos \theta_3 \omega_3 - r_4 \cos \theta_4 \omega_4 = 0 \quad (7)$$

Because we have solved the mechanism previously, the only unknowns are ω_3 and ω_4 for each given configuration $(\theta_2, \theta_3, \theta_4)$.

2.1.3 Numerical velocity calculation

If we have plotted a coupler curve with a sufficient number of points, (x_i, y_i) , we can also estimate the velocity vector at each point as

$$\vec{V}_i = (1/\delta t)[(x_{i+1} - x_i), (y_{i+1} - y_i)]. \quad (8)$$

2.1.4 Adding the overall machine velocity

Note that in each case above, we have obtained the velocity of a point with respect to the boat, i.e., ${}^b\vec{V}_i$. Meanwhile, the boat is moving with some velocity, ${}^w\vec{V}_b$ with respect to the world. So the absolute velocity of our point with respect to the world at each instant is: ${}^w\vec{V}_i = {}^w\vec{V}_b + {}^b\vec{V}_i$. The same vector addition applies to walking machines.

2.2 Obtaining the Force

The other part of the power equation is the force, $\vec{F}(t)$. Estimating the ground reaction force requires some judgement and, ideally, a multi-axis force plate to measure normal and tangential forces, as used in the Royal Society article on pentapedal Kangaroos. With luck, we may have such a device for the first time in ME112 thanks to some adapted tactile sensor technology from Alice Wu and Isabel Gueble. Without a force plate, one can still make some useful estimates.

2.2.1 Static force balance

Because the machine is not steadily accelerating upward or downward, we know the average vertical force has to be mg . Knowing the weight of your machine, the approximate center of mass location, and how many feet are on the ground, you can get an idea of the static load on each foot at each time. We also know that the horizontal force at each foot is limited by friction: $f_{ti} \leq \mu f_{ni}$ where f_{ti} and f_{ni} are the tangential and normal forces at the i 'th contact point.

2.2.2 Dynamic force balance

As feet impact the ground and impart some upward and forward thrust to the body, the dynamic forces will be higher than the average forces.

You can get a rough idea of the dynamics by looking at some slow-motion video of your machine. Often we find that the center of mass follows a roughly sinusoidal motion path. For example, suppose the center of mass rises and lowers by 0.01m per stride with a stride frequency of 2.0Hz. Then the vertical COM motion is approximated by $y = Y \sin(\omega t)$ where $Y = 0.01\text{m}$ and $\omega = 2 \cdot 2\pi$ radians/second. The vertical acceleration is $\ddot{y} = -\omega^2 Y \sin(\omega t) = 1.6\text{m/s}^2$. This is small compared to gravity.

However, it's likely that the motion is less smooth than a perfect sinusoid. For example, suppose there are some "higher order" terms and the vertical motion is best approximated by a short Fourier series:

$$y = Y_1 \sin(\omega t) + Y_2 \sin(2\omega t) \quad (9)$$

The acceleration is now

$$\ddot{y} = -\omega^2 Y_1 \sin(\omega t) - 4\omega^2 Y_2 \sin(2\omega t). \quad (10)$$

If $Y_1 = Y_2 = 0.01\text{m}$, and we have a stride period of 2Hz, the peak acceleration is now

$$\ddot{y}_{max} = 5 \cdot (4\pi)^2 \cdot 0.01 \quad (11)$$

which comes to about 7.9m/s^2 . In fact, a conservative rule of thumb for dynamic loading is to assume that the total acceleration could be as much as $2g$, effectively doubling the static weight. Again, the horizontal force will be limited by $f_{ti} \leq \mu f_{ni}$. In practice, it will usually be considerably smaller unless the feet are actually slipping.

3 Virtual Work

Recall that the companion to power ($\vec{F} \cdot \vec{V}$) is *virtual work*: $\delta W = \vec{F} \cdot \delta \vec{X}$. Conveniently, if we have a trajectory of points (p_{xi}, p_{yi}) , we can easily compute $(\delta x_i, \delta y_i)$ so that

$$\delta w_i = \begin{bmatrix} f_{xi} & f_{yi} \end{bmatrix} \begin{bmatrix} \delta x_i \\ \delta y_i \end{bmatrix}. \quad (12)$$

Finally, suppose we have two feet driven by a single motor that rotates with a torque, τ_2 , and an angle, θ_2 . Then for each time step, δt , we have:

$$\delta w = \tau_2 \cdot \delta \theta_2 = \begin{bmatrix} f_{1x} & f_{1y} \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta y_1 \end{bmatrix} + \begin{bmatrix} f_{2x} & f_{2y} \end{bmatrix} \begin{bmatrix} \delta x_2 \\ \delta y_2 \end{bmatrix} \quad (13)$$

where the (1,2) subscripts now correspond to the first and second foot, respectively.